Remembering Where and When

COMP3431 Robot Software Architectures

So far ...

- Robots we have discussed so far have simple sensors
- Do not build complex models of the world
- Do not require memory

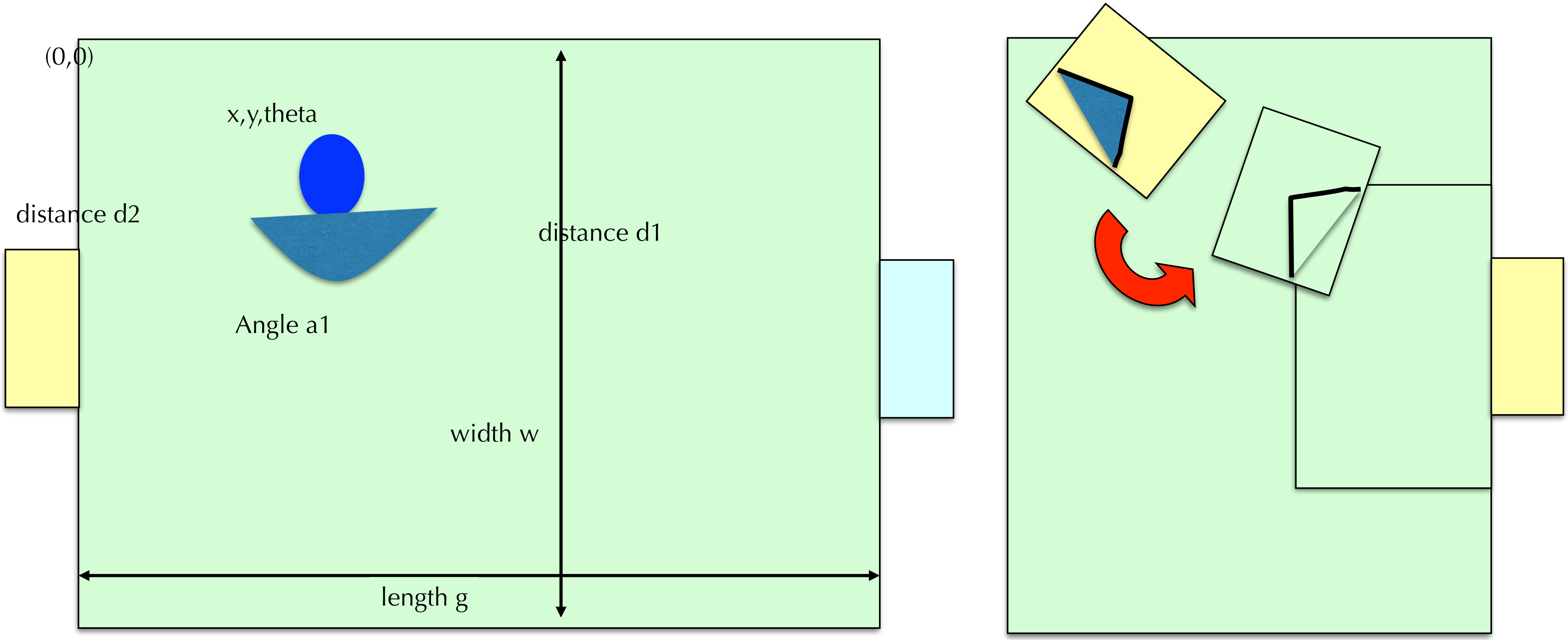
This time ...

- We look at the robot equivalent of spatiotemporal memory

Probabilistic Robotics

Localisation with Landmarks

Localisation with Edges



Errors

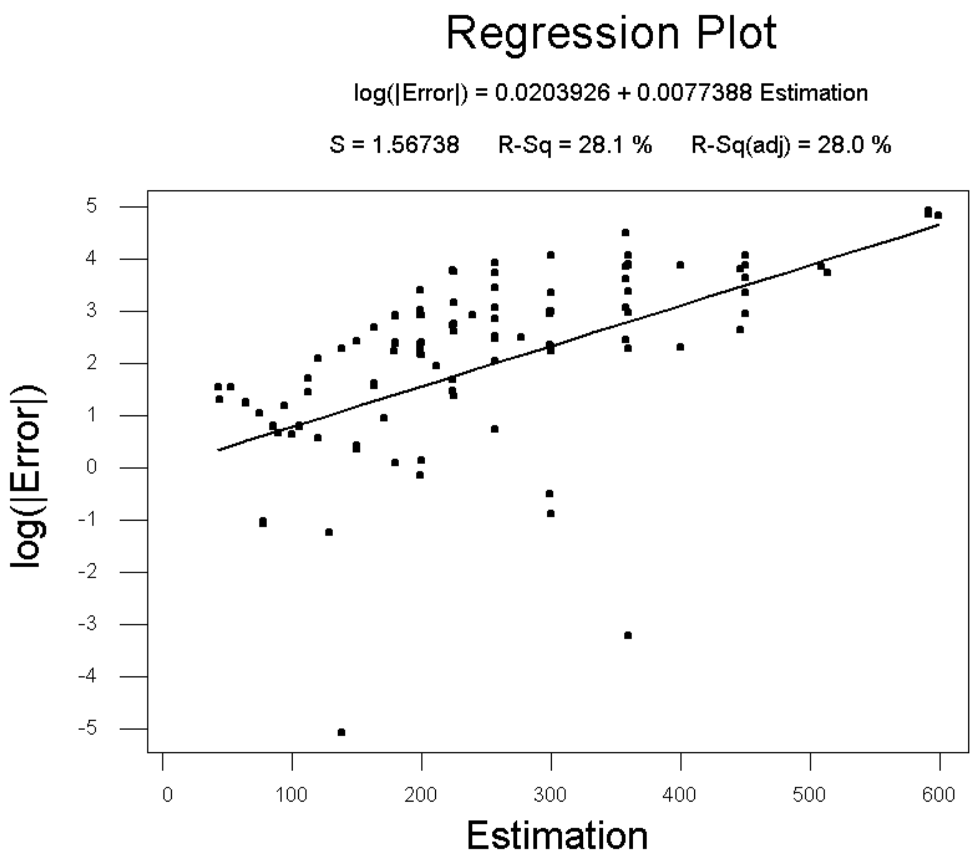
- Measurement errors
    - sensors are never 100% accurate
- Process errors
    - actions never do exactly what they’re supposed to

Example

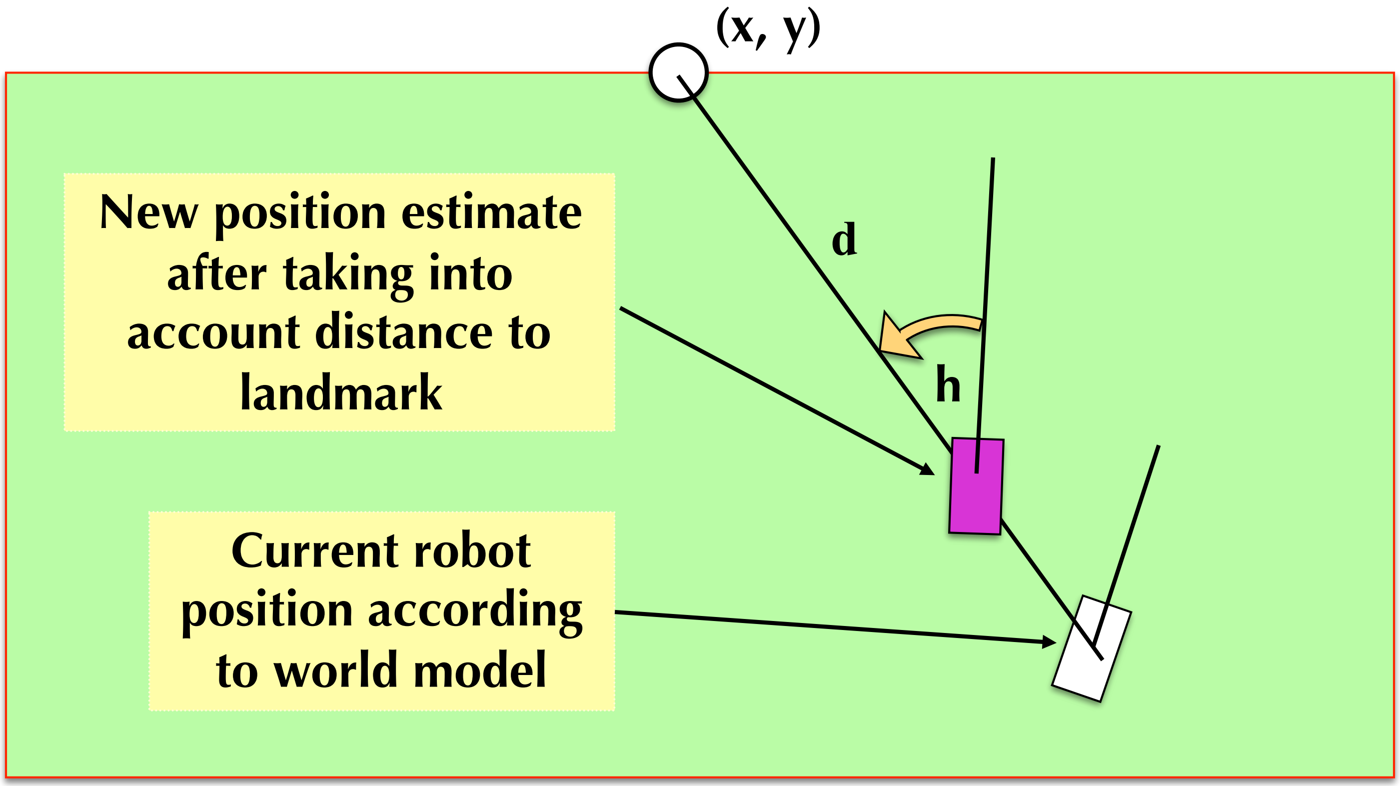
- Estimating distance to the ball

Measurement Error

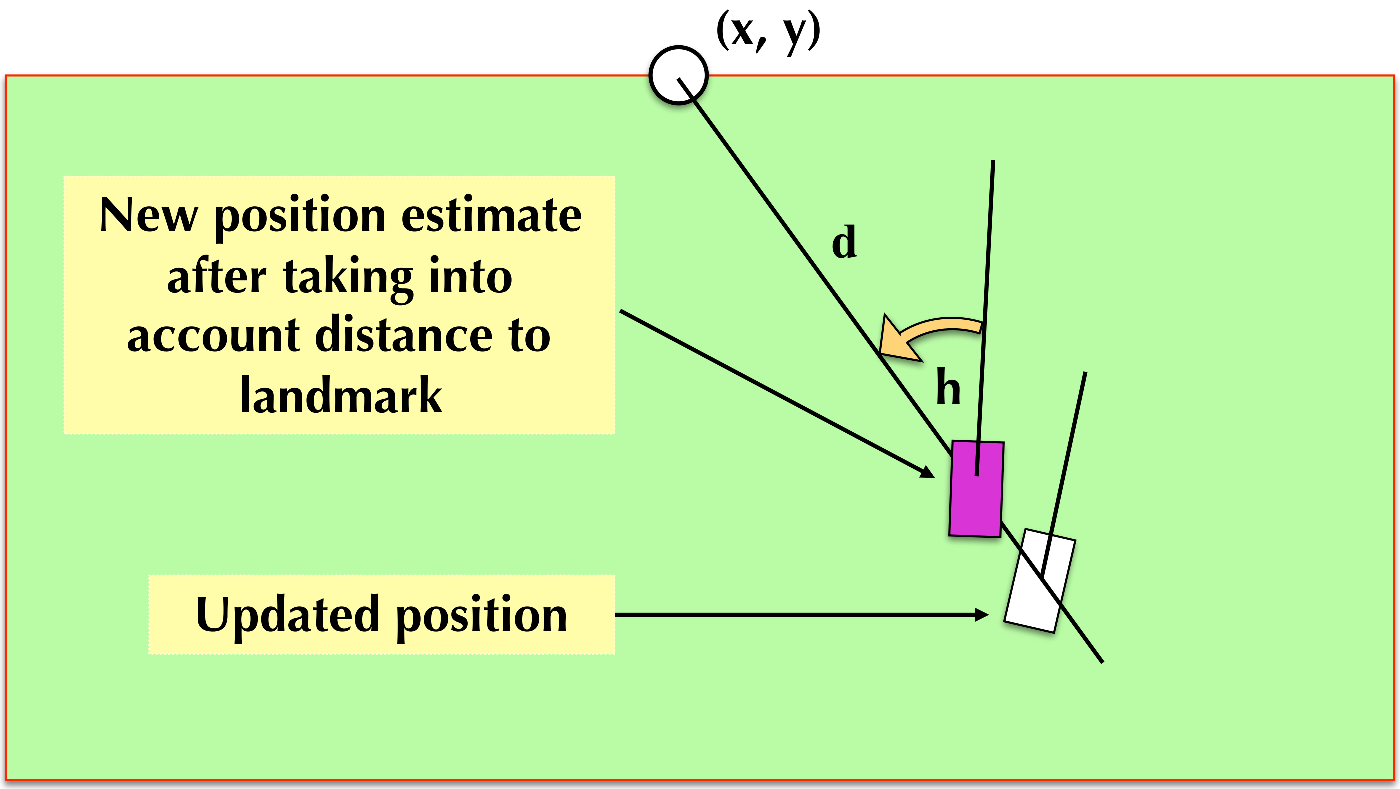
Experiments determine errors in distance estimation



Localisation with Landmarks



Localisation with Landmarks



Conditional Probability

$P(E|H) \equiv$ the probability of observing E if H is true

Bayes' Rule

- Probability I have a cold if you hear me cough

$$P(cold \mid cough) \propto P(cold) \times P(cough \mid cold)$$

- I.e. if we know the prior probability that I have a cold (without any evidence) and I know that a cold causes a cough, with some probability, then we can calculate the posterior probability

Probabilistic Inference

- Make inferences using probabilities
- Based on Bayes' rule:

$$P(H \mid E) \propto P(H) \times P(E \mid H)$$

or

$$bel(x_t) \propto bel(x_{t-1}) \times prob(observation)$$

Updating State Estimate

- The Kalman filter is commonly used to update the estimate of the robot's state
- Two phases:
    1. Prediction (Process Update)
        - predicts where the robot will be after performing an action
    2. Correction (Observation Update)
        - use observations to correct prediction
- What follows is only a sketch of the Kalman filter
    - It's nowhere near the complete algorithm

Simplification

- Only one measurement and action
- When there are more, must account for all interactions
- Math becomes more complex
  - Scalar variables are replaced by matrices

Process Update (Simplified)

$$\bar{x} \leftarrow x + u$$

$$var_x \leftarrow var_x + Q$$

- $\bar{x}$ is the *predicted* new state after action, u
- Update variance with process noise, Q
  - errors accumulate

Measurement Update

- Move *position estimate* toward *measurement estimate* but proportional to *error estimates*

$$x \leftarrow \bar{x} + K \times \left(z - \bar{x} \right)$$

- z is the measured state

- new state estimate is predication plus difference between prediction and measurement, proportional to confidence in measurement

Measurement Update

$$x \leftarrow \bar{x} + K \times \left(z - \bar{x}\right)$$

$$\text{if } k = 1$$

$$x \leftarrow \bar{x} + z - \bar{x}$$

$$x \leftarrow z$$

- update x by the difference in the measured value, z, and the expected value, x, scaled by how much we trust the observation
- If measurement is certain, new state becomes measured state
- otherwise, make change proportional to difference between measurement and prediction

Measurement Update (Simplified)

$$K \leftarrow \frac{var_x}{var_x + R}$$

$$var_x \leftarrow (1 - K) \times var_x$$

$$x \leftarrow \bar{x} + K \times (z - \bar{x})$$

- As measurement error R approaches 0, actual measurement z is trusted more and more, while predicted measurement $\bar{x}$ is trusted less and less.
- As state error estimate var_x approaches 0, actual measurement z is trusted less and less, while predicted measurement $\bar{x}$ is trusted more and more.

Robot estimating state of door

Initial beliefs

- Door can be in one of two states, open or closed
- Represented by state variable, X
- Initially, X has equal probability of being open or closed

$$bel\left(X_0 = open\right) = 0.5$$
$$bel\left(X_0 = closed\right) = 0.5$$

Measurement Noise

- Specify the probability of sensor given correct answer
- Z is the measurement, X is the actual value

$$p\left(Z_t = \text{sense\_open} \middle| X_t = \text{is\_open}\right) = 0.6$$
$$p\left(Z_t = \text{sense\_closed} \middle| X_t = \text{is\_open}\right) = 0.4$$
$$p\left(Z_t = \text{sense\_open} \middle| X_t = \text{is\_closed}\right) = 0.2$$
$$p\left(Z_t = \text{sense\_closed} \middle| X_t = \text{is\_closed}\right) = 0.8$$

- Only 0.2 error probability when door is closed
- but 0.4 error probability when door is open

Process Noise

- Robot uses its manipulator to push door open
- If already open, door stays open
- If closed, robot has 0.8 chance that door will be open after a push

$$p\left(X_t = \text{is\_open} \middle| U_t = \text{push}, X_{t-1} = \text{is\_open}\right) = 1$$

$$p\left(X_t = \text{is\_closed} \middle| U_t = \text{push}, X_{t-1} = \text{is\_open}\right) = 0$$

$$p\left(X_t = \text{is\_open} \middle| U_t = \text{push}, X_{t-1} = \text{is\_closed}\right) = 0.8$$

$$p\left(X_t = \text{is\_closed} \middle| U_t = \text{push}, X_{t-1} = \text{is\_closed}\right) = 0.2$$

Process Noise

- The robot may do nothing
- World does not change

$$p\left(X_t = \text{is\_open} \middle| U_t = \text{do\_nothing}, X_{t-1} = \text{is\_open}\right) = 1$$
$$p\left(X_t = \text{is\_closed} \middle| U_t = \text{do\_nothing}, X_{t-1} = \text{is\_open}\right) = 0$$
$$p\left(X_t = \text{is\_open} \middle| U_t = \text{do\_nothing}, X_{t-1} = \text{is\_closed}\right) = 0$$
$$p\left(X_t = \text{is\_closed} \middle| U_t = \text{do\_nothing}, X_{t-1} = \text{is\_closed}\right) = 1$$

Probabilistic Robotics

- Belief in a state variable x at time t is its probability at t given all past measurements and actions:

$$bel(x_t) = p(x_t \mid z_{1..t}, u_{1..t})$$

- Belief after action u_t but before observation z_t, i.e. after prediction but before correction:

$$\overline{bel}(x_t) = p(x_t \mid z_{1..t-1}, u_{1..t})$$

Bayes' Rule

- Don't have to use entire history
- Use Bayes' Rule

$$bel(x_t) \propto prob(observation) \times bel(x_{t-1})$$

- Belief is a probability distribution over state variable
- Update must sum probabilities of outcomes of actions for each possible value

Example

- If door is open and robot pushes, what is the outcome?

- If door is open and robot does nothing, what is the outcome?

- If door is closed and robot pushes, what is the outcome?

- If door is closed and robot does nothing, what is the outcome?

Bayes Filter

For all state variables

Predict value after the next action

Update the value based on the next measurement

$$\text{forall } x_t \text{ do}$$

$$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1})\, bel(x_{t-1})\, dx_{t-1}$$

$$bel(x_t) = \eta\, p(z_t \mid x_t)\, \overline{bel}(x_t)$$

Prediction for x_t is the sum of predictions for each value of x_t

Update prediction by last observation

η is a normalising factor to keep probabilities in 0 .. 1.

Example

- At $t = 1$ the robot takes no action but senses an open door
  - u_1 = do_nothing
  - z_1 = sense_open

Integral becomes a sum because values of x are discrete

$$
\begin{aligned}
\overline{bel}(x_1) &= \int p(x_1 | u_1, x_0)\, bel(x_0)\, dx_0 \\
&= \sum_{x_0} p(x_1 | u_1, x_0)\, bel(x_0) \\
&= p(X_1 | U_t = \text{do\_nothing}, X_o = \text{is\_open})\, bel(X_0 = \text{is\_open}) \\
&\quad + p(X_1 | U_t = \text{do\_nothing}, X_o = \text{is\_closed})\, bel(X_0 = \text{is\_closed})
\end{aligned}
$$

Example

Substitute values for X_1

$$\overline{bel}(x_1 = \text{is\_open})$$
$$= p(X_1 = \text{is\_open} | U_t = \text{do\_nothing}, X_o = \text{is\_open})\, bel(X_0 = \text{is\_open})$$
$$+ p(X_1 = \text{is\_open} | U_t = \text{do\_nothing}, X_o = \text{is\_closed})\, bel(X_0 = \text{is\_closed})$$
$$= 1 \times 0.5 + 0 \times 0.5$$
$$= 0.5$$

$$\overline{bel}(x_1 = \text{is\_closed})$$
$$= p(X_1 = \text{is\_closed} | U_t = \text{do\_nothing}, X_o = \text{is\_open})\, bel(X_0 = \text{is\_open})$$
$$+ p(X_1 = \text{is\_closed} | U_t = \text{do\_nothing}, X_o = \text{is\_closed})\, bel(X_0 = \text{is\_closed})$$
$$= 0 \times 0.5 + 1 \times 0.5$$
$$= 0.5$$

Measurement Update

$$bel(x_1) = \eta\, p(z_1 = \text{sense\_open} \mid x_1)\, \overline{bel}(x_1)$$

$$\begin{aligned} bel(x_1 = \text{is\_open}) &= \eta\, p(z_1 = \text{sense\_open} \mid x_1 = \text{is\_open})\, \overline{bel}(x_1 = \text{is\_open}) \\ &= \eta \times 0.6 \times 0.5 \\ &= \eta \times 0.3 \end{aligned}$$

$$\begin{aligned} bel(x_1 = \text{is\_closed}) &= \eta\, p(z_1 = \text{sense\_open} \mid x_1 = \text{is\_closed})\, \overline{bel}(x_1 = \text{is\_closed}) \\ &= \eta \times 0.2 \times 0.5 \\ &= \eta \times 0.1 \end{aligned}$$

$$\eta = \frac{1}{0.3 + 0.1} = 2.5$$

$$\begin{aligned} bel(x_1 = \text{is\_open}) &= 0.75 \\ bel(x_1 = \text{is\_closed}) &= 0.25 \end{aligned}$$

Normalise to ensure that probabilities add up to 1

Iterate for more actions

If the next action is **push** and the measurement is **sense_open**:

$$\overline{bel}(x_1 = \text{is\_open}) \quad = 1 \times 0.75 + 0.8 \times 0.25 \quad = 0.95$$

$$\overline{bel}(x_1 = \text{is\_closed}) = 0 \times 0.75 + 0.2 \times 0.25 \quad = 0.05$$

and

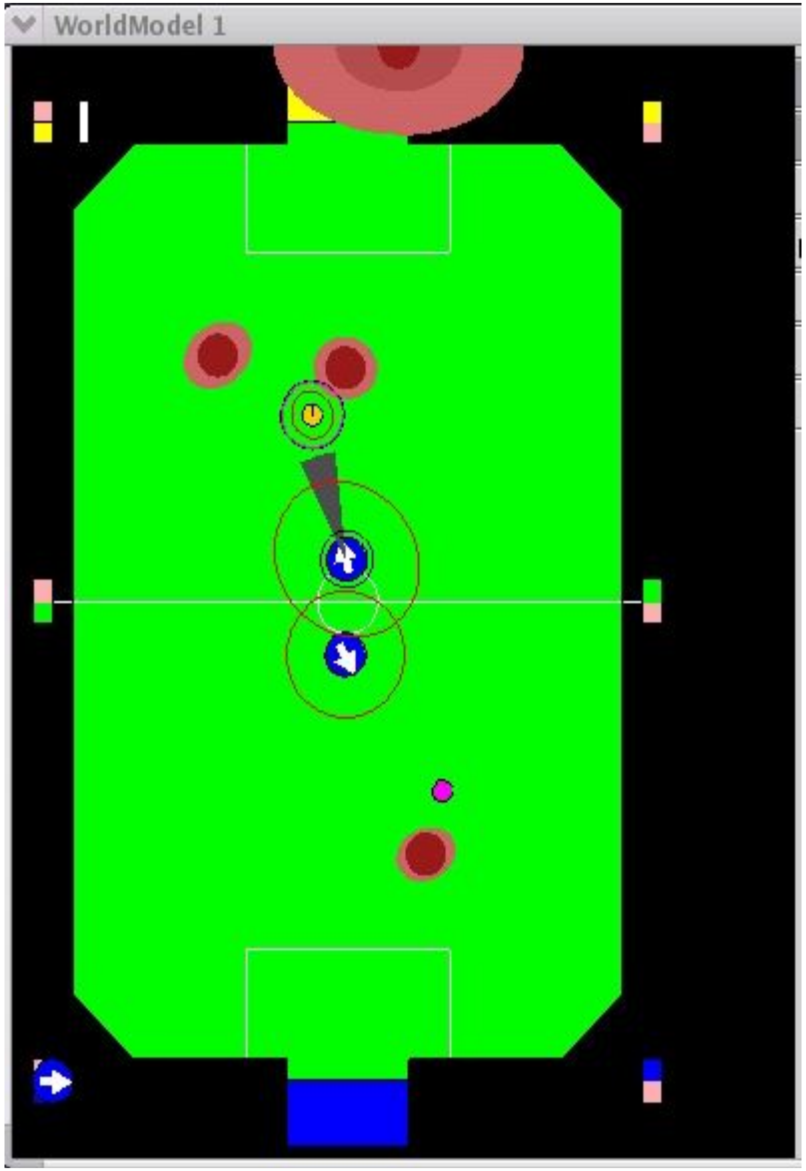
$$bel(x_1 = \text{is\_open}) \quad = \eta \times 0.6 \times 0.95 \quad \approx 0.983$$

$$bel(x_1 = \text{is\_closed}) = \eta \times 0.2 \times 0.05 \quad \approx 0.017$$

Position Tracking

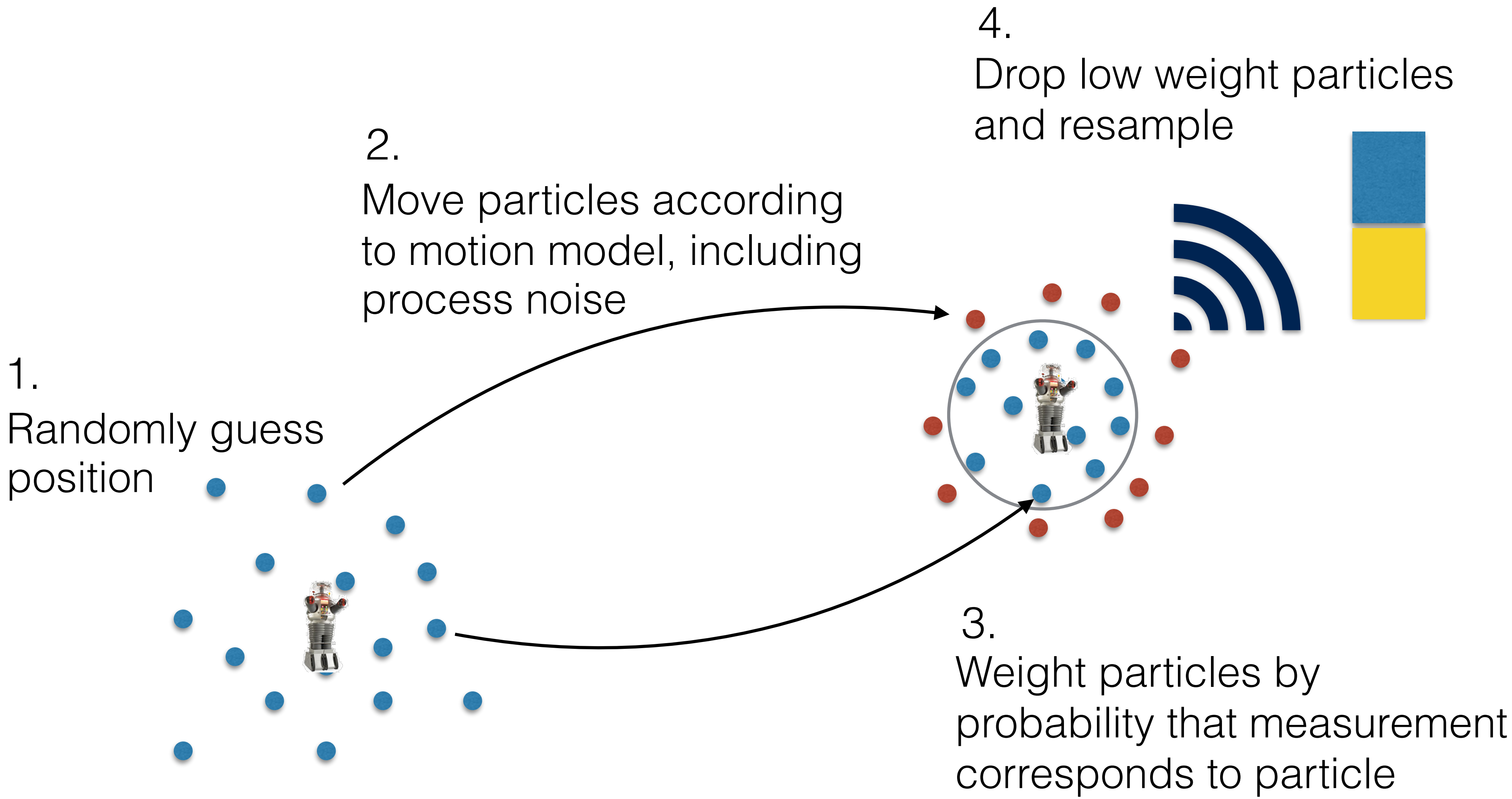
- Robot moves
  - Predict new position based on what motor actions are expected to do
- Measure
  - Uses sensors to estimate motion
- Update position estimate (often a Kalman Filter)

RoboCup Localisation



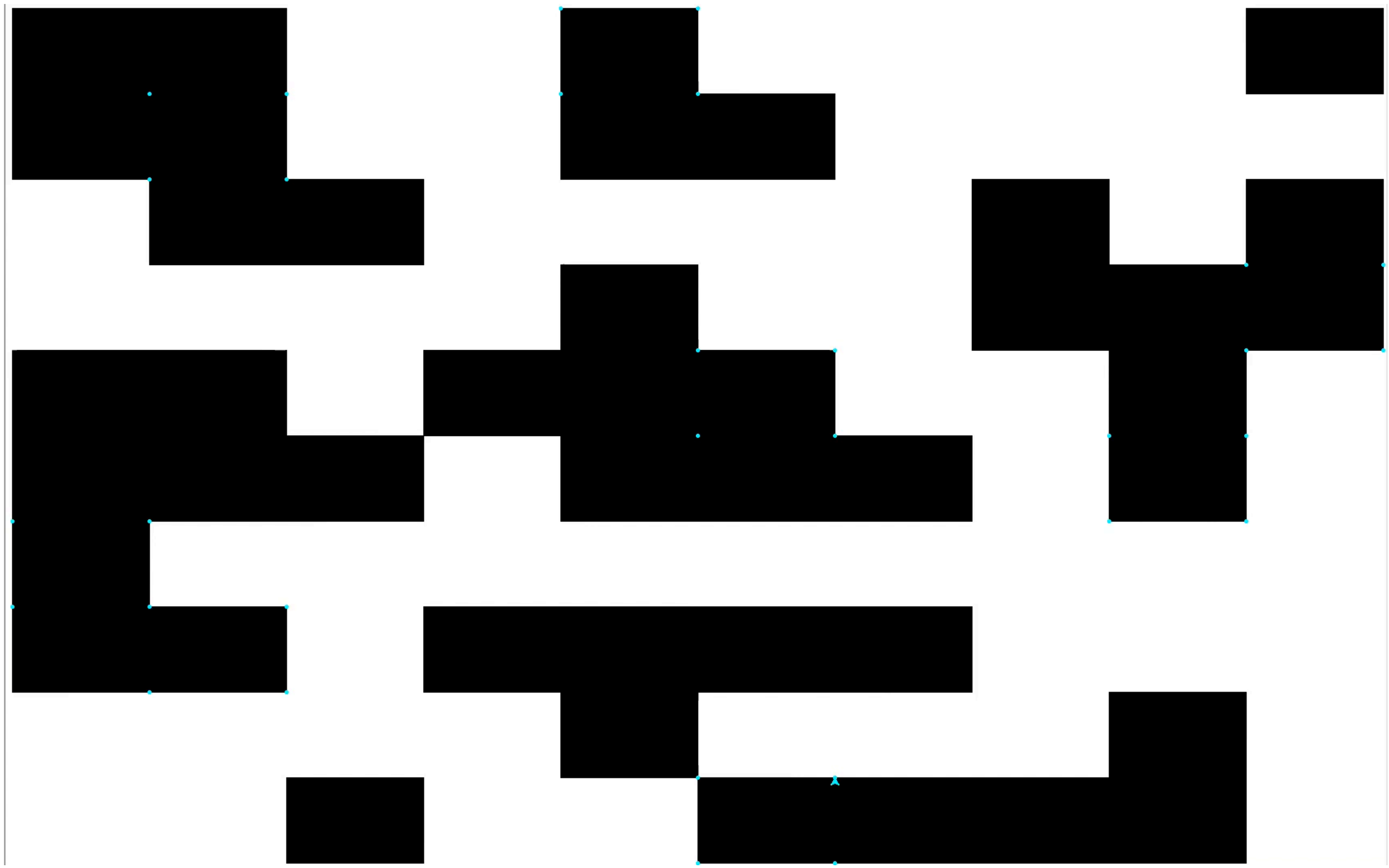
- Estimates of robot and ball positions include variance (or error)
- Robot has errors in
  - x
  - y
  - heading
- Robot variance is shown as an ellipse and sector

Particle Filter



https://github.com/mjl/particle_filter_demo

Particle Filter



Particle Filter

$ParticleFilter(\chi_{t-1}, u_t, z_t)$

$\bar{\chi}_t = \chi_t = \emptyset$

$for\ m = 1\ to\ M:$

$\quad sample\ x_t^{[m]} \sim p\left(x_t \mid u_t, x_{t-1}^{[m]}\right)$

$\quad w_t^{[m]} = p\left(z_t \mid x_t^{[m]}\right)$

$\quad \bar{\chi}_t = \bar{\chi}_t + \left\langle x_t^{[m]}, w_t^{[m]} \right\rangle$

$for\ m = 1\ to\ M:$

$\quad draw\ i\ with\ probability\ \alpha\ w_t^{[i]}$

$\quad add\ x_t^{[i]}\ to\ \chi_t$

$return\ \chi_t$

Each particle is a hypothesis of what the state is at time t

Sample from the state transition distribution

Weight is probability of measurement for particle

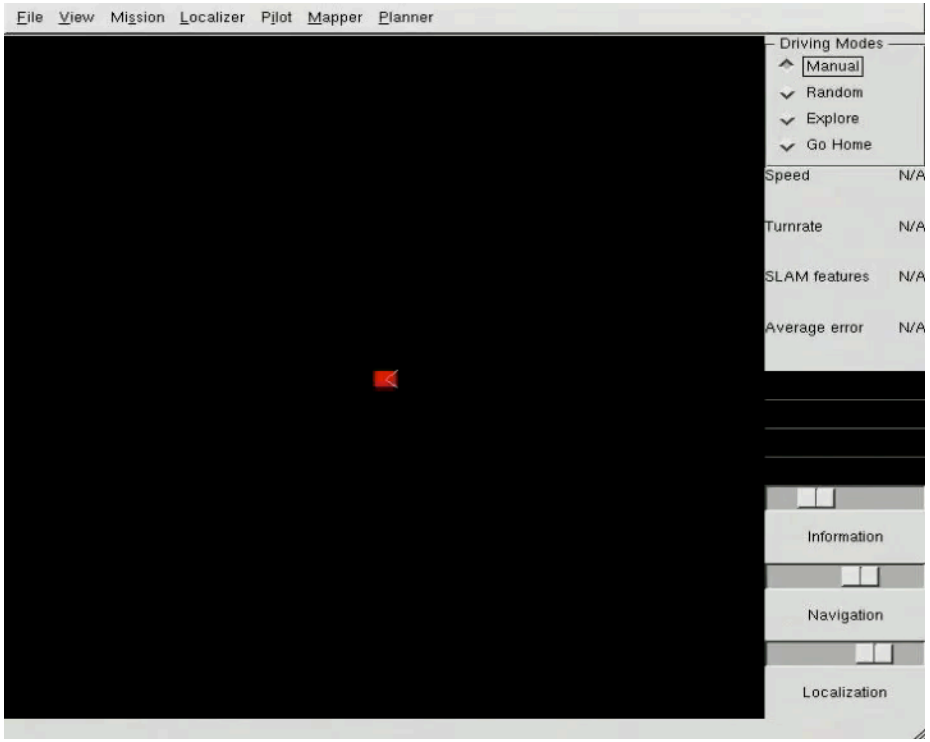
Resample by weight

Measurement Errors

- Position tracking usually uses wheel encoders to estimate motion
- Unreliable in rescue robots
- We use lasers and RGB-D cameras
- Estimate motion from difference in successive scans

Simultaneous Localisation and Mapping (SLAM)

- Depth sensors also give distance to objects
- Similar estimation methods can be used to update map



Loop Closure (Full SLAM)

- Position tracking alone will accumulate errors
- If the robot recognise a landmark that it has seen before
  - it can correct drift by updating estimate based on measurement of landmark
- Error correction is back-propagated

Probabilistic Robotics

- Position tracking, mapping, localisation
- How confident are we that a robot arm has gripped an object?
- Is what I'm seeing really a ball or is it a cylinder, end-on?
- Is the ground ahead a flat, traversable surface or is it the surface of a deep lake?
- If I drive into that obstacle, what are the chances that it's a bush that I can go over or it's a boulder that I'll crash into?
- How confident is my autonomous car in detecting pedestrians?